WHAT IS CLAIMED IS:

1. An interface between a master and one or more slave modules comprising:

a master;

a slave having a set of addressable registers including egress mailbox registers, ingress mailbox registers, and indirect access address registers;

a direct memory access (DMA) engine coupled to the master by a first bus and to the slave by a second bus, with the second bus comprising:

a set of bi-directional data lines for transmitting data between the slave and the DMA engine;

a set of master address lines for transmitting address data from the DMA engine to the slave;

a master data strobe for strobing data;

a master read/write signal for indicating whether data is to be read from or written to the slave;

a set of slave select signals for selecting one of a plurality of slaves connected to the second bus;

a slave wait signal asserted by a slave to delay a data transfer;

a slave reset signal,

a clock output signal, and

a clock input signal;

where the DMA engine performs direct data transfers to the slave by asserting a slave select signal to the slave and transferring data over the set of bi-directional data lines to the slave egress or ingress data registers and performs indirect data transfers to slave memory by writing address data over the set of bi-directional data lines to the indirect address register of the slave and where the slave utilizes its own memory map and the address data to transfer data between a location indicated by the address data and the DMA engine.

2. The interface of claim 1 where the DMA engine negotiates with a slave to implement either an asynchronous, synchronous, or source synchronous data transfer.

3. The interface of claim 1 where the DMA engine negotiates with all slaves during reset to determine the maximum bus width available to transfer data.

4. The interface of claim 1 where:

the slave includes status register and message signal interrupt (MSI) register; and

where the slave asserts a bit in the status register to indicate it is ready for a transaction and where the DMA engine asserts a bit in the MSI register to indicate when a transaction is complete.

5. A method for allowing a DMA engine to provide access to a plurality of slave devices to multiple masters, the protocol, implemented by hardware and software on the DMA engine, the master, and the slave devices, comprising the steps of:

to implement a direct message transfer to a slave device:

accessing a slave status register to read a direct message ready status bit which is set when the slave is ready to transfer data;

transferring message data using the DMA engine and a slave mailbox register if the direct message ready status bit is set;

setting an message transfer complete status interrupt at the slave to indicate when the transfer of the message is complete; and

to implement an indirect data transfer to the memory space of a slave device:

accessing a slave status register to read an indirect message ready status bit which is set when the slave is ready to transfer data;

transferring address data using the DMA engine and slave indirect address mailbox register if the indirect message ready status bit is set;

setting an indirect transfer message interrupt bit at the slave to initiate the indirect transfer;

transferring message data between the DMA engine and slave mailbox registers if the indirect message ready status bit is set, where the slave utilizes its own memory map and the address data to transfer data between a location indicated by the address data and the DMA engine; and

setting an message transfer complete status interrupt at the slave to indicate when the transfer of the message is complete.

6. The method of claim 5 further comprising the step of:

negotiating with all the slaves to implement either an asynchronous, synchronous, or source synchronous data transfer.

7. The method of claim 5 further comprising the step of:

starting the bus upon reset at a fixed bus-width and then negotiating with all the slaves to implement acceptable bus bit-width.

8. A system for allowing a DMA engine to provide access to a plurality of slave devices to multiple masters, the protocol, implemented by hardware and software on the DMA engine, the master, and the slave devices, said system comprising:

means for implementing a direct message transfer to a slave device including:

means for accessing a slave status register to read a direct message ready status bit which is set when the slave is ready to transfer data;

means for transferring message data using the DMA engine and a slave mailbox register if the direct message ready status bit is set;

means for setting an message transfer complete status interrupt at the slave to indicate when the transfer of the message is complete; and

means for implement an indirect data transfer to the memory space of a slave device including:

means for accessing a slave status register to read an indirect message ready status bit which is set when the slave is ready to transfer data;

means for transferring address data using the DMA engine and slave indirect address mailbox register if the indirect message ready status bit is set;

means for setting an indirect transfer message interrupt bit at the slave to initiate the indirect transfer;

means for transferring message data between the DMA engine and slave mailbox registers if the indirect message ready status bit is set, where the slave utilizes its own memory map and the address data to transfer data between a location indicated by the address data and the DMA engine; and

means for setting an message transfer complete status interrupt at the slave to indicate when the transfer of the message is complete.

9. The system of claim 8 further comprising:

means for negotiating with all the slaves to implement either an asynchronous, synchronous, or source synchronous data transfer.

10. The system of claim 8 further comprising:

means for starting the bus upon reset at a fixed bus-width and then negotiating with all the slaves to implement acceptable bus bit-width.